

Available online at www.sciencedirect.com**SciVerse ScienceDirect**

Procedia - Social and Behavioral Sciences 59 (2012) 287 – 296

Procedia
Social and Behavioral Sciences

UKM Teaching and Learning Congress 2011

Major problems in basic programming that influence student performance

Masura Rahmat*, Shahrina Shahrani, Rodziah Latih, Noor Faezah Mohd Yatim, Noor Faridatul Ainun Zainal & Rohizah Ab Rahman

Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia

Abstract

Programming is an important basic skill for computer science students. However, most students found it is hard to learn and score a good mark. This paper discusses the study to identify problems and causes faced by programming students. The study was conducted at the Faculty of Information Science and Technology, UKM. Data was collected from 179 students who answered on-line questionnaire. The study identified several major problems; reference materials, lectures and lab session approaches, problem solving ability, time management and self-confidence. The findings will be used to improve the courses and thus ensures the realization of course outcomes, decrease the failure rates and increase students' interest in programming.

© 2011 Published by Elsevier Ltd. Selection and/or peer reviewed under responsibility of the UKM Teaching and Learning Congress 2011 Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/3.0/).

Keywords : Method of delivery; programming; student problems; students performance; reference materials

1. Introduction

Programming is part of the basic curriculum for computer science and information technology. The ability to understand and implement the programming language is important skill for the students to acquire. Faculty of Information Science and Technology (FTSM), UKM offers the C++ programming subject (TK1914) and Program Design and Problem Solving subject (TK1924) to all students as a compulsory course. These subjects are pre-requisite for the other programming courses offered at the third and subsequently semester. Both of these subjects are taught using C++ programming language and conducted by lecture (theory) and practical exercise in the laboratory.

This paper discusses the study to identify the students' problems in the subject of programming and causes of those problems. The study also investigates whether the causes of the problems manipulate student performance in this subject. Student performance is measured from their examination grade. The focus of this study is mainly to answer the following questions:

1. What type of the reference material used by students and how did they use it?
2. Do students prepare themselves before lectures?
3. What is the student's perception towards teaching methods used?

* Corresponding author. Tel.: +6-03- 6761-5449; fax: +6-03-8925-6732
E-mail address: masura@ftsm.ukm.my

4. Does the student do revision, and what methods do they use?
5. Do students do assignments on their own or sharing it with others, and why?
6. What programming skills are acquired according to the students?

2. Literature Review

Computer programming is a subject that involved skill in designing algorithm, writing program, understanding the syntax as well as the logic of the program. Programming is among the hardest subject to learn due to some reasons (Daly, 1999) (Jenkins, 2002). We group these reasons into three categories; the programming subject itself, the students and the teachers. We discuss each of these factors in the following section.

2.1 Subject

The programming subject itself is a main reason that makes the courses difficult because it is new to most of the students. Programming is a hierarchical skill where students acquire a basic skill before advance level. For example, students need to learn the basic syntax and then the semantic, structure, and style gradually. However, during the class, the lecturer will continue teaching without waiting the student to fully understand the topic. Otherwise, the course syllabus cannot be finished. Programming is not only a hierarchical skill but also a multiple skills. Programming is a process start with translating an algorithm into program code. The hardest part is to interpret the requirement specification into algorithm. The right algorithm will produce the right program. Therefore, students are required to skilled in all processes; design the algorithm, translate the algorithm into program code and write the program code with the correct syntax.

Generally programming subject is offered level by level; beginner class is in first semester, intermediate class is in the second and third semester and advance class is in subsequent semester. Students who fail in the examination needs to repeat the course however those who passed will go to the next level. Unfortunately, for some students even though they passed the examination but they still lack of confident in programming. When they took the next course, they will face a problem where they need to learn a new thing as well as improving their programming skill.

The programming language used as a teaching introductory programming course also is a problem. Generally, the aim of the programming course is to teach student on how to program. Programming language is just a medium used to achieve that aim. However, most programming language used for teaching is not actually design for teaching but instead for the industry. Therefore, it is for sure not appropriate language.

Most students believe that programming is a difficult subject to understand and score a good mark. This view is passed to new students and thus gives bad impression that this subject is the hardest subject.

Programming is a new subject for many students. We cannot expect students to be skilled in programming within 14 weeks. However it may took 10 years for someone to be skilled programmer (Winslow, 1996).

2.2 Students

Students' interest in the subject is also important. Learning programming will be boring and difficult if they are not interested in the subject. The way the teachers conduct the class also can be a factor that makes student not interested with the subject. Different students have different types of motivation. Some students motivated to learn programming because of their interest while the others were motivated by their parents or friends.

Students' learning style is also different for each student. Some like discussion with friends while other prefers to study alone. Whatever their styles are but the most important is the way they thinking. This is because learn to program involved a different way of thinking (Mayer, Dyck et al., 1986).

2.3 Teachers

Teachers play a prominent role in delivering the knowledge efficiently to the students (Gomes and Mendes, 2007). Teachers are responsible to explain to the students clearly and suggest to them the solution for their problems. Learning environment like size of the class, time slot for the class, and duration of the class also can affect students' motivation (Jenkins, 2001). These can be a problem to the students if the teachers fail to handle the situation besides the various learning styles of the students.

3. Methodology

A set of questionnaires were developed as an instrument of this research which includes six main sections: (i) Section A: Respondent profile, (ii) Section B: Perception towards programming, (iii) Section C: Lecture materials and

reference, (iv)Section D: Teaching method, (v)Section E: Revision method and assignments, and (vi) Section F: Programming skills.

Section A covers information on respondent's background and their programming experience. Section B asks about the perception before (pre-course) and after (post-course) students takes the programming subjects. Section C is related to the use of materials for the lectures and other references. Teaching methods and course delivery by lecturers and teachers are questioned in Section D. Section E focuses on revision methods, handling of assignment and time management. Finally, section F assesses student performance in terms of grades, types of motivation, and perception of the programming skills they acquired. However, this paper only focuses on the problem potentially faced by students as well as the source of problems in Section C, D and E. More importantly, this paper will also discuss the relationship between the problems and student performance either directly or indirectly. Student performance is measured by grades achieved for the course which is gathered from section F.

This questionnaire was distributed online to all students taking the programming subject in the second semester of 2010/2011 session. The reliability of the questionnaire items is very good with Cronbach alpha reliability of the scale of 0.919. Descriptive and inferential analysis was performed using Statistical Package for the Social Science (SPSS) software version 19.

4. Result

4.1 Demographic Analysis

From the total of 179 respondents, 65.4% of them are female and 34.6% from them are male. This is probably due to the higher number of female than male students in the general population of this faculty.

Based on the total of respondents as shown in Figure 1, the percentage of Malay students are 60.9% followed by Chinese (31.3%), Indian (2.8%), and while the percentage for other native students and non-citizens are 1.7% and 3.4% respectively. Figure.1 shows the percentage of students based on races.

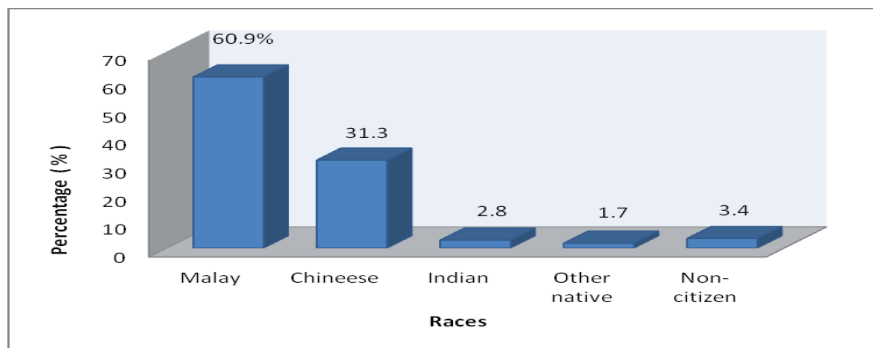


Figure 1. Percentage of respondents according to race

There are six programs in FTSM; (i) Intelligent Systems (TC), (ii) Computer Science (TK), (iii) Information Science (TP), (iv) Multimedia Studies (TH), (v) Industrial Computing (TR), and (vi) Science and Management System (TS). Figure.2 shows the distribution of students by program, and from that we can identify the highest percentage of involvement comes from TK and TH program.

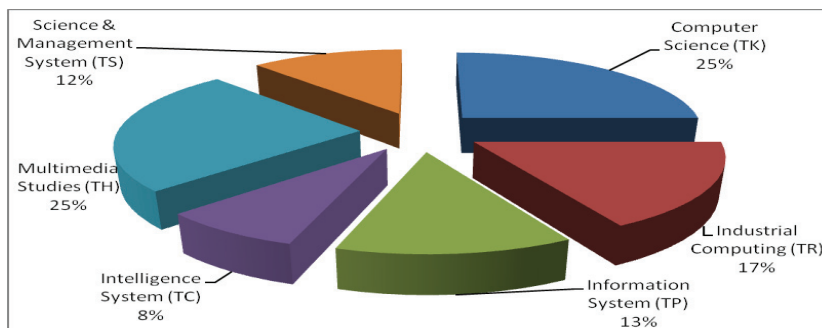


Figure 2. Percentage of respondents according to Program.

Student performance is measured by grades and classified into four main groups; excellent, good, moderate and weak, as shown in Table 1. Students who belong to an excellent group performance is the students who obtained grade A or A- in the subject of programming that is represented 24% of the total students. Students with a grade of B+, B and B- are categorized in good performance students while students who earn grades of C+, C, and C- is the moderate of the student groups. 6.7% of students classified as weak with grade D+, D and E. Achievement of this category will be used to see whether there are significant differences between each of the problems that have been analyzed with student performance.

Table 1. Student performance according to groups

Performance	Grades	Frequency	Percentage (%)
Excellent	A, A-	43	24
Good	B+, B, B-	79	44.1
Moderate	C+, C, C-	45	25.1
Weak	D+, D, E	12	6.7

The Kruskal-Wallis test was used for inferential analysis to see whether there are significant differences between the problems and student performance. This test was chosen because the data is abnormal and scattered groups tested are more than two. Inferense analysis is related to the issue that has been mentioned previously.

4.2 What type of the reference material used by students and how did they use it?

78% of total respondents said that the lecture notes provided by lecturers are not enough. Table 2 shows the frequency of various forms of reference used by students. Based on the table, 66.5% of the respondents use the reference books recommended by lecturers while 53.6% refer to other reading materials such as tutorials available from the Internet. However, a high percentage of respondents (93.9%) uses and depend on the lecture notes, while 55.3% of them uses the additional reference materials.

Table 2. Frequency of use of reference material

Form of reference	Frequency	Percentage (%)
Notes / slides from lecturer	168	93.9
Notes from senior	64	35.8
Additional reference books	99	55.3
Other reading materials (such as tutorials) from the Internet)	96	53.6
Reference books recommended by lecturers	119	66.5

Only 35.8% of the respondents refer to notes from their seniors. Based on the analysis using the Kruskal-Wallis test, we found significant differences of frequency of senior notes usage ($H(3) = 8018$, $p = 0046$) between the different performance groups, where the excellent group has the mean rank value (101.71), good (93.12), moderate (77.89) and weak (72.92). Notes from seniors refers to personal notes taken during lectures, lab sessions or revisions based on their own understanding of the subject. This shows that extra reference material, especially notes from seniors used by the students may help them to obtain good grades for the subject.

Based on this analysis, there are 33.5% of respondents do not use the reference books recommended by lecturers, while 44.7% do not use other additional reference books. When asked about the reasons of why the reference books are not used, 65% of respondents have chosen the answer that stated that the prices of the books are expensive. Another popular reason of why reference books are not used is that the exercises in these books are not provided with answers (63%). Figure.3 shows in detail the percentage of each possible reason for not using the reference books.

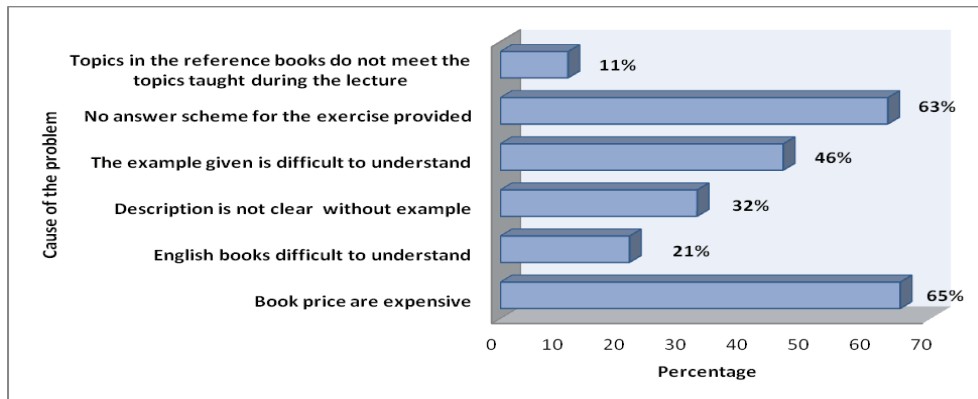


Figure 3. Reasons for not using reference books.

There was significant difference of mean rank between different performance groups for the following reasons:

1. Price of reference book is expensive ($H(3) = 9.134$, $p = 0.028$)
2. Books written in English are difficult to understand ($H(3) = 8.018$, $p = 0.046$)
3. Descriptions in the book is not clear, and no examples provided ($H(3) = 15.495$, $p = 0.001$)
4. Examples given the book are difficult to understand ($H(3) = 11.305$, $p = 0.010$)
5. No answer scheme for the exercises provided ($H(3) = 13.116$, $p = 0.004$)

Mean rank values for the each performance group of each reason are shown in Table 3.

Table 3. Mean rank for the reason for not using reference books

Cause of the problem	Performance Group	Frequency	Mean Rank
Price of reference book is expensive	Excellent	43	101.19
	Good	79	93.68
	Moderate	45	76.26
	Weak	12	77.25
Books written in English are difficult to understand	Excellent	43	79.83
	Good	79	88.49
	Moderate	45	101.33
	Weak	12	93.88
Description in the book is not clear, and no examples provided	Excellent	43	75.57
	Good	79	91.59
	Moderate	45	106.74
	Weak	12	68.46
Examples given in the book are difficult to understand	Excellent	43	70.40
	Good	79	93.95
	Moderate	45	99.21
	Weak	12	99.71
No answer scheme for the exercise provided	Excellent	43	69.38
	Good	79	97.44
	Moderate	45	93.67
	Weak	12	101.13

The Excellent group has chosen the expensive price of the book as the most popular reason for not using reference books. Students from the Moderate group however, blames it more on the difficulties in understanding books written in English language and unclear descriptions and unavailability of examples. The Weak group has chosen the difficulties of understanding the examples provided and unavailability of answers as their most popular reasons for not using reference books. This indicates that weak students are prone to depend on the answers to facilitate their understanding of a problem in programming.

4.3 Do students prepare themselves before lectures?

Overall results of the analysis found that 63.2% of respondents often download notes/slides before lectures, although only 24.1% always read and understand the notes/slides downloaded. Majority of students (74.8%), always note down the key points or additional information during lectures but only 48.6% have the initiative to read and try to understand the notes/slides after the lecture. A total of 45.3% of respondents doubled their efforts by reading reference books related to lecture topics.

4.4 What is the student's perception towards teaching methods used?

Two teaching methods are used in this course: (i) lectures delivered by lecturers, and (ii) laboratory exercises conducted and guided by IT teachers. 65.9% of respondents agreed that the appropriate time for lectures is between 1-2 hours, while 55.3% expressed the same number of hours for laboratory sessions. This indicates that the time allocated for lectures and lab sessions determined by the faculty administration is adequate and in conformity with the respondents.

4.4.1 Lectures by Lecturers

Based on analysis, 42.5% of respondents agreed that examples provided during lectures are sufficient while 43% indicated that the slides used by lecturers are attractive and meet the students' needs. Both percentages are the lowest compared with five other question items as shown in Figure.4.

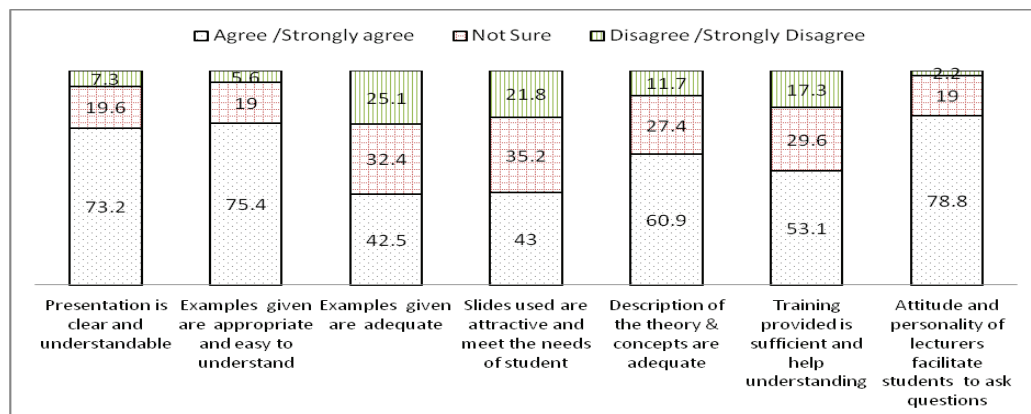


Figure 4. Students perception towards lectures handling method

There is a statically significant difference between the different performance group ($H(3) = 9.143$, $p = 0.027$) for the adequacy of example given during lectures with the mean rank for the Excellent (107.70), Good (88.98), Moderate (79.46) and Weak (72.83) performance groups. This shows that students categorized in Excellent group felt that examples given during lectures are sufficient. A possible explanation to this reaction is that these students may have their own initiatives to try other question or examples from other sources of references.

4.4.2 Lab Sessions by IT Teachers

A total of 66.5% of students agree and strongly agree the assistance given by IT teacher in identifying the errors is sufficient. Figure.5 shows the percentage of other issues relating to laboratory sessions.

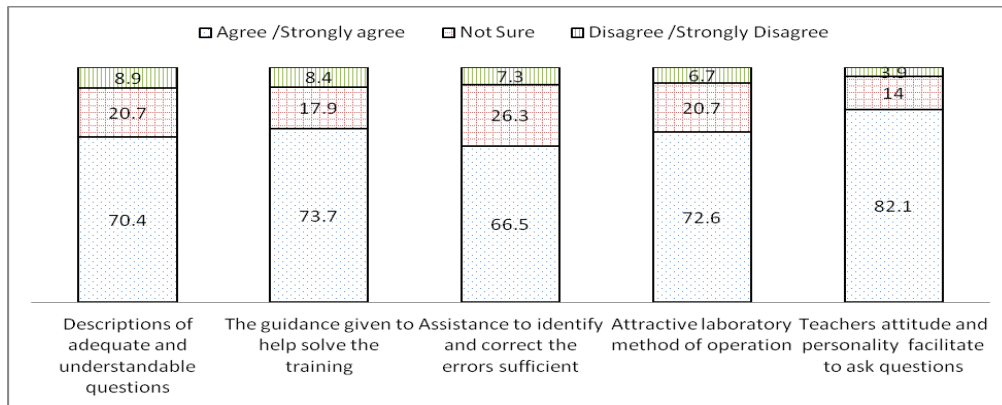


Figure 5. Students perception towards lab handling method

There are significant differences between different performance groups for the following issues relating to laboratory sessions:

- Assistance to identify and correct errors is sufficient ($H(3) = 15.420$, $p = 0.001$) with a mean rank of the Excellent (106.97), Good (91.48), Moderate (69.10) and Weak (97.83).
- Attitude and personality of teachers encourage students to ask questions ($H(3) = 8.146$, $p = 0.043$) with the value of the mean rank for the Excellent (98.31), Good (86.45), Moderate (81.01) and Weak (117.29).

The results showed that the assistance provided by IT teachers to identify and correct the errors is sufficient for students with excellent performance. Students with weak performance agreed that the attitude and personality of IT teachers encourage them to ask questions. However, this does not seem to help them to achieve good grades.

4.5 Does the student do revision, and what methods do they use?

A total of 78.8% of respondents reported that they always or every time do the exercises required by lecturers. However, only 37.4% reported that they always or every time do the recommended exercises (not required) by the lecturers or teachers. Only 27.4% of respondents reported that they always and every time do exercises on their own initiatives. Figure 6 shows the percentages of different type of revisions done by respondents.

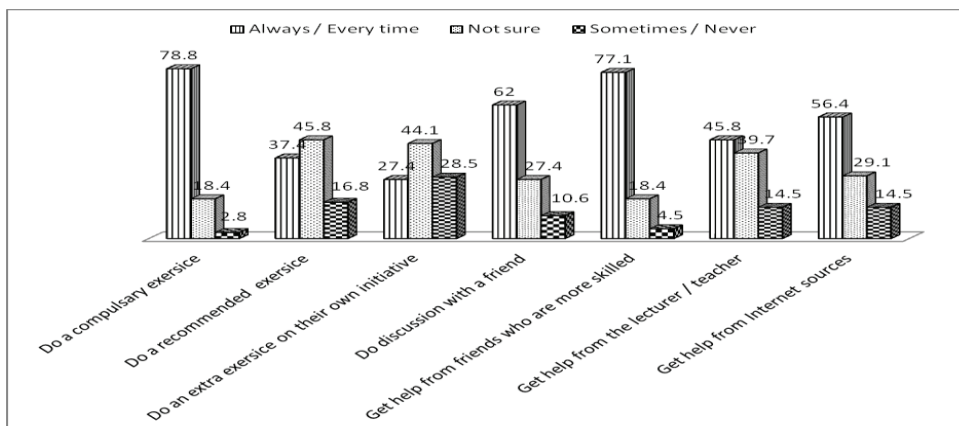


Figure 6. Students revision method

There are significant differences between the different performance groups for the following revision methods:

- Do additional exercises on their own initiatives ($H(3) = 8.578$, $p = 0.035$) with the mean rank for Excellent s (108.50), Good (86.68), Moderate (80.12) and Weak (82.58).

- b) Seek assistance from lecturers or teachers if the topic is not understood ($H(3) = 15.451$, $p = 0.001$) with the mean rank for the Excellent (107.71), Good (91.70), Moderate (80.47) and Weak (51.08).

The results showed that Excellent students made the effort to do additional exercises on their own initiatives. They also always seek the assistance of lecturers if they do not understand a particular topic. When asked about the average time spent in a week to do revision for the programming subject, 55.9% of respondents which is the majority, answered 2-5 hours, while 22.9% of respondents estimated the time spent in a week to do revision is not more than 2 hours.

4.6 Do students do assignments on their own or sharing it with others, and why?

A total of 53.1% of respondents reported that they can complete assignments on their own with a little help from friends or lecturers, while 17.9% of respondents always and every time share the answers with friends but did not understand the answers submitted. Although the amount is small but the problem of sharing answers or copying is a problem that should be addressed.

There are significant differences between the different performance groups for the following methods of completing and submitting assignments:

- Complete assignments by sharing the answers and fully understand the answers submitted ($H(3) = 18.084$, $p = 0.000$)
- Complete assignments by sharing the answers but do not fully understand the answers submitted ($H(3) = 8.088$, $p = 0.044$)
- Complete assignments by sharing the answers and do not understand the answers submitted at all ($H(3) = 12.109$, $p = 0.007$)

Mean rank values for each of the above method are listed in Table 4.

Table 4. Mean rank for methods of assignment problem solving

Problem	Performance group	Frequency	Mean Rank
Complete assignments by sharing the answers and fully understand the answers submitted	Excellent	43	112.14
	Good	79	89.73
	Moderate	45	77.89
	Weak	12	57.88
Complete assignments by sharing the answers but do not fully understand the answers submitted	Excellent	43	71.31
	Good	79	96.54
	Moderate	45	94.69
	Weak	12	96.33
Complete assignments by sharing the answers and do not understand the answers submitted at all	Excellent	43	69.35
	Good	79	90.94
	Moderate	45	103.27
	Weak	12	108.08

Students who performed weakly often share the answers but not understand the answers submitted. Students who performed excellently however, fully understand the answers submitted even shared. Based on the questionnaire, there are three factors chosen by respondents to be the source of the problem that leads to answers sharing. 62% of respondents reported that they regularly and each time feel the question is too difficult and requires discussion with a friend. A total of 58.1% of respondents often find the assignment question is too difficult to solve individually, and 50.3% reported that they often try but fail to solve the problems themselves. Apart from these three reasons, respondents have also reported of occasions where they are not familiar with programming syntax, do not understand the error messages, do not know how correct the errors, do not understand the theories/concepts related to the problem, do not understand how to apply the related theories/concepts explained during lectures on the assignments and do not have enough time to complete the assignments.

There are significant differences between performance groups for the following reasons of the sharing the answers:

- The question is too difficult to understand and needs discussion with a friend ($H(3) = 18.865$, $p = 0.000$)
- The question is too difficult to solve individually ($H(3) = 18.938$, $p = 0.000$)

- c) Have tried answering the question but failed ($H(3) = 15.867, p = 0.001$)
- d) Not familiar with the related programming syntax ($H(3) = 31.594, p = 0.000$)
- e) Do not understand the error messages given by the compiler ($H(3) = 14.716, p = 0.0002$)
- f) Do not know how to correct the errors indicated by the compiler ($H(3) = 10.154, p = 0.017$)
- g) Do not understand the theories/concepts related to the question ($H(3) = 10.017, p = 0.018$)
- h) Do not know how to apply the concepts explained during lectures on the question ($H(3) = 17.259, p = 0.001$)

Mean rank values for each of the above reasons are listed in Table 5.

Table 5. Mean rank for the reasons for sharing answer

Reason	Performance group	Frequency	Mean Rank
The question is too difficult to understand and needs discussion with a friend	Excellent	43	64.93
	Good	79	92.95
	Moderate	45	107.44
	Weak	12	95.00
The question is too difficult to solve individually	Excellent	43	66.12
	Good	79	90.27
	Moderate	45	109.87
	Weak	12	99.29
Have tried answering the question but failed	Excellent	43	66.45
	Good	79	93.11
	Moderate	45	107.13
	Weak	12	89.63
Not familiar with the related programming syntax	Excellent	43	60.87
	Good	79	89.50
	Moderate	45	119.13
	Weak	12	88.42
Do not understand the error messages given by the compiler	Excellent	43	65.93
	Good	79	94.15
	Moderate	45	104.10
	Weak	12	96.08
Do not know how to correct the errors indicated by the compiler	Excellent	43	72.70
	Good	79	90.54
	Moderate	45	105.53
	Weak	12	90.17
Do not understand the theories/concepts related to the question	Excellent	43	71.58
	Good	79	90.90
	Moderate	45	103.20
	Weak	12	100.58
Do not know how to apply the concepts explained during lectures on the question	Excellent	43	63.66
	Good	79	96.30
	Moderate	45	103.29
	Weak	12	93.04

4.7 What programming skills are acquired according to the students?

Majority of respondents (62.6%) agreed or strongly agreed that they have the skills to understand what is being done by a program and able to predict the output of the program. A total of 54.2% of respondents agreed that they can understand the error messages given by the compiler.

There are significant differences between different performance groups for the following perceived programming skills:

- a) Able to understand and predict the output to be produced ($H(3) = 11.747, p = 0.008$) with the mean rank value for the Excellent (107.87), Good (89.36), Moderate (80.27) and Weak (66.67)
- b) Able to correct self-written programs based on the error messages given by compiler ($H(3) = 11.947, p = 0.008$) with the mean rank value for the Excellent students (111.20), Good (84.37), Moderate (84.33) and Weak (72.33)

- c) Able to correct programs written by others based on the error messages given by compiler ($H(3) = 15,899$, $p = 0.001$) with the mean rank value for the Excellent students (115.17), Good (83.37), Moderate (79.49) and Weak (82.83)
- d) Able to write a segment of program that performs a particular process ($H(3) = 19.196$, $p = 0.000$) with the mean rank value of the Excellent students (113.19), Good (89.78), Moderate (68.39) and Weak (89.38)
- e) Able to write a complete program ($H(3) = 23.675$, $p = 0.000$) with the mean rank value of the Excellent students (114.06), Good (92.57), Moderate (63.90) and Weak (84.75)

5. Discussion

Based on the results of the analysis performed on six key questions to investigate major problems in programming, the study found that students do not have high level of confidence to perform individual tasks and depend on each other. Students initiative is also very minimal because most of them, particularly students with moderate and weak performance always hope for help from other sources such as the answer scheme from reference books, and assistance from lecturers, teachers and friends to help them solve particular exercises or tasks. More importantly than that, student's effort in obtaining materials for lectures and laboratory sessions is still at a minimum level. The majority of students still depend entirely on notes or slides prepared by lecturers even though they are aware that these notes are not enough to help their understanding. There are also students who use other reference sources but the numbers are too small. The study also found students who performed excellently use various alternatives to increase their knowledge by getting notes from senior friends and other references from the Internet.

In addition, with lots of time spent on lectures or lab sessions for all the subjects taken, it is very difficult for students to do revisions and extra exercises to enhance their programming skills. Large number of students in one class may also affect the focus of the instructor and students. The situation makes it difficult for lecturers and teachers to identify students who are weak in basic programming skills. Among the proposed improvements that can be done is to use the pair programming method as an alternative, in order to help students solve their programming problems more constructively. This method can also help students write programs with higher level of confidence than doing them individually. Indirectly, the problem of sharing answers can be reduced. Use of forums as a medium should be more encouraged, so that students do not feel embarrassed to ask the lecturers or teachers face to face. This method could be helpful for moderate and weak students because they sometimes feel unsure of what kind of the question should they ask.

6. Conclusion

In conclusion, these findings has identified the problems exist in programming subjects among students and their causes. Improvements in the method of conducting these courses must be a high priority and should be enhanced from time to time. Lecturers and teachers can use alternative methods to attract students and help increase the interest s in the world of programming. Students should be more prepared with acquiring notes before class, brief reading the topics to be taught, noting down the important points during lectures, asking the lecturers if they do not understand the theory, asking the teachers if the syntax of program if difficult to understand during laboratory sessions, doing additional exercises and finding other sources or references. Students are also expected to improve their learning style and be more competitive to place themselves in a group of students with excellent performance.

Acknowledgement

We would like to thank Universiti Kebangsaan Malaysia for providing the research grant (UKM-PTS-2011-049).

References

- Daly, C. (1999). RoboProf and an introductory computer programming course. *SIGCSE Bull.* 31(3): 155-158.
- Gomes, A. and Mendes, A. J. (2007.) Learning to program - difficulties and solutions. *International Conference on Engineering Education – ICEE* 2007, Coimbra, Portugal.
- Jenkins, T. (2001). The motivation of students of programming. *Proceedings of the 6th annual conference on Innovation and technology in computer science education*. Canterbury, United Kingdom, ACM: 53-56.
- Jenkins, T. (2002.) On the difficulty of learning to program. In *3rd annual Conference of LTSN-ICS*.
- Mayer, R.E. , Dyck, J. L., et al., (1986). Learning to program and learning to think: what's the connection? *Commun. ACM* 29(7): 605-610.
- Winslow, L.E. (1996). Programming pedagogy - a psychological overview. *SIGCSE Bull.* 28(3): 17-22.